7

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/579,357	05/26/2000	Michael G. Burke	Y0R9-2000-0038US1-(8728-3	7577

7590 04/07/2004

Frank Chau Esq
F Chau & Associates LLP
1900 Hempstead Turnpike Suite 501
East Meadow, NY 11554

EXAMINER	
RAMPURIA, SATISH	
ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 04/07/2004

3

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/579,357	**Applicant(s)** BURKE ET AL.	7
	Examiner Satish S. Rampuria	**Art Unit** 2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *03* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 May 2000*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date *8/28/2000*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is in response to the application filled on May 26, 2000.

2. Claims 1-29 are pending.

Information Disclosure Statement

3. An initialed and dated copy of Applicant's IDS form 1449, Paper No. 2, received on 8/28/00 is attached to the instant Office action.

Priority

4. An application in which the benefits of an earlier application are desired must contain a specific reference to the prior application(s) in the first sentence of the specification of in an application data sheet (37 CFR 1.78(a)(2) and (a)(5)). The specific reference to any prior nonprovisional application must include the relationship (i.e., continuation, divisional, or continuation-in-part) between the applications except when the reference is to a prior application of a CPA assigned the same application number.

Specification

5. The disclosure is objected to because it contains an embedded hyperlink and/or other form of browser-executable code on page 19, line 23, page 20, line 1, page 53, and lines 22-23. Applicant is required to delete the embedded hyperlink and/or other form of browser-executable code. See MPEP § 608.01.

6. The abstract of the disclosure is objected to because contains more than 150 words. Correction is required. See MPEP § 608.01(b).

Claim objections

7. Claim 28 is objected to because of the following informalities: claim 28 is dependent on claim 25. It appears that this claim should be dependent on claim 26. It is assumed in this office action that claim 28 is dependent on claim 26.

Appropriate correction is required.

Claim Rejections - 35 USC § 112, second paragraph

8. Claims 1, 12, 15, 26, and 29 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Clarification and/or correction are required.

Regarding, claim 1, on lines 9, 14, and 18, the limitation, "capable of" is unclear as to during the execution of the program classes within set A is capable of loading classes outside set A or it is loading the classes outside set A.

Claim 12, has the similar limitation to those in claim 1 with respect to "capable of", recited on the lines 8 and 13.

Claim 15, has the similar limitation to those in claim 1 with respect to "capable of", recited on the lines 6 and 11.

Claim 26, has the similar limitation to those in claim 1 with respect to "capable of", recited on page 65 and 66, lines 3 and 7.

Claim 29, has the similar limitation to those in claim 1 with respect to "capable of", recited on the lines 14 and 19.

The rejection of the base claim is necessarily incorporated into the dependent claims.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1, 2, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over to US Patent No. 6,546,551 to Sweeney et al., hereinafter called Sweeney, in view of Advanced JAVA 2nd edition published in 1999 by Berg, hereinafter called Berg,.

Per claim 1:

Sweeney disclose:

- A method for analyzing an object oriented program (col. 11, line 56 "A method for analyzing an object-oriented program") ***that support dynamic class loading*** (col. 2, lines 4-7 " Dynamic loading … a class with a specified name, after… instances of… class… created"), ***the program comprising a plurality of objects*** (col. 11, lines 63-64 "plurality of components comprises classes"), ***each class belonging to at least one class*** (col. 9, line 59 "dynamically load class M, create an object of type M")

- ***identifying a set A of classes in the program*** (col. 8, line 29 "identifying a set I of classes")

- ***identifying a first set of method calls*** (col. 8, line 27 "identifying a set M of methods… calls") ***belonging to the classes in the set A*** (col. 8, lines 37-38 "the data identifying the set R is initialized to identify a set of initially reachable methods (such as an application's main method)"), ***during execution of the program*** (col. 12, lines 2-3 "during… execution of the program"), ***calling only methods belonging to a class within the set A*** (col. 12, lines 27-29 "components is satisfied if at least one component in the first set of components")

- ***identifying a second set of method calls*** (col. 8, line 27 "identifying a set M of methods… calls") ***belonging to the classes in the set A*** (col. 8, lines 37-38 "the data identifying the set R is initialized to identify a set of initially reachable methods (such as an application's main method)"),

- ***storing data*** (col. 3, lines 37-38 "The memory") ***that identifies the first and the second set of method calls for subsequent use*** (col. 8, line 27 "identifying a set M of methods to which virtual calls")

Sweeney does not explicitly disclose each of the classes within the set A is capable of during execution of the program, causing loading of a class outside of the set A.

However, Berg discloses in an analogous computer system class loader loads a class from different domains at runtime (page 249, 1st paragraph "A class loader loads classes from one or more domains… codesources, typically single codesource").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of loading a class from outside of a class at runtime as taught by Berg into the method of analyzing the of a program as taught by Sweeney. The modification would be obvious because of one of ordinary skill in the art would be motivated load the class from within the class which provide powerful JAVA platform to install software components at runtime as taught by Berg (page 249, paragraph 1).

Per claim 2:

The rejection of claim 1 is incorporated, and further, Sweeney disclose:

- reporting the stored data to a user (col. 11, lines 46-47 "The identified component(s) are then reported to the user via the display device")

Claim 26 is the product claim corresponding to method claim 1 and rejected under the same rational set forth in connection with the rejection of claim 1 above.

11. Claims 3 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg in view of US Patent No. 6,622,300 to Krishnaswamy et al., hereinafter called Krishnaswamy.

Per claim 3:

The rejection of claim 1 is incorporated, and further, neither Sweeney nor Berg discloses optimizing the program, based upon the stored data.

However, Krishnaswamy discloses in an analogous computer system optimizing program based on the data (col. 2, lines 10-12 "optimization program then optimizes… computer program based on … data").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of optimizing the program based on the data as taught by Krishnaswamy in the method of analyzing the program as taught by the combination system of Sweeney and Berg. The modification would be obvious because of one of ordinary skill in the art would be motivated to optimize the program which shared across the computer system as taught by Krishnaswamy (col. 3, lines 25-33)

Claim 27 is the product claim corresponding to method claim 3 and rejected under the same rational set forth in connection with the rejection of claim 3 above.

12. Claims 4, 6, 7, 8, 11, and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg, in view of US Patent No. 6,631,518 to Bortnikov et al., hereinafter called Bortnikov.

Per claims 4 and 11:

The rejection of claim 3, is incorporated, and further, Sweeney disclose:

- ***for a given method call*** (col. 7, line 28 "For a given call… methods") ***in the second set*** (col. 6, line 38 "second set of component")
- ***adding an optimization to the program*** (Abstract, "whole-program optimizations")

Neither Sweeney nor Berg discloses validity and triggering of an optimization program.

However, Bortnikov disclose in an analogous computer system validity and execution of optimization (col. 2, lines 62-64 "optimization mechanism that optimizes using any available valid procedure"). It is inherent to execute the program for optimization for optimization mechanism.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of valid optimization of a program as taught by Bortnikov into the combination system as taught by Sweeney, Berg, and Kwong. The modification would be obvious because of one of ordinary skill in the art would be motivated to validate optimization program for optimization of the program to efficiently run the system as suggested by Bortnikov (col. 3, lines 21-36).

Per claim 6, 7, and 8:

The rejection of claim 4 is incorporated, and further, Sweeney disclose:

- ***adding a bit*** (col. 9, line 60 "assign the value '10'... integer field x") ***in a class table*** (col. 9, line 56 "small class library") ***corresponding to a given class and setting the bit to a first predefined value when the given class is within the set A*** (col. 10, lines 27-45 "initial value... sets are... instructions in the... method... are scanned... contains virtual calls... a direct call... an initiation of class C... data identifying ... is added to the data identifying set M" and col. 8, lines 64-67 "Steps... perform... new classes are added to set...")

Claim 28 is the product claim corresponding to method claim 4 and rejected under the same rational set forth in connection with the rejection of claim 4 above.

13. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg in view of US Patent No. 6,289,506 to Kwong et al., hereinafter called Kwong.

Per claim 5:

Neither Sweeney nor Berg discloses performing static analysis in identifying the first and second set.

However, Kwong discloses in an analogous computer system performing static analysis (col. 8, lines 27-28 "perform static and dynamic analysis of the program's performance").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of performing static analysis to a program as taught by Kwong into the method of analyzing the program as taught by the combination system of Sweeney and Berg. The modification would be obvious because of one of ordinary skill in the art would be motivated to perform the static analysis of the program to get the optimized code as taught by Kwong (col. 2, lines 20-34)

14. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg in view of Dynamic Class Loading in the Java Virtual Machine, published in 1998 by Liang et al., hereinafter called Liang.

Per claim 9:

The rejection of claim 1 is incorporated, and further, neither Sweeney nor Berg disclose classes in the set A are identified by a user.

However, Liang disclose in an analogous computer system class loaded and specify which class to be loaded by the user (page 1, section "Introduction-- ...A user-defined class loader can, for example, specify the remote location from which the classes are loaded").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of loading user defined class as taught by Liang into the method of analyzing the program as taught by the combination system of Sweeney and Berg. The modification would be obvious because of one of ordinary skill in the art would be motivated to load a class by user to maintain type safety as suggested by Liang (page 1, Abstract).

15. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg, Bortnikov in view of Marmot: An optimization Compiler for JAVA, published in 1999 by Fitzgerald et al., hereinafter called Fitzgerald.

Per claim 10:

The rejection of claim 4 is incorporated, and further, Sweeney disclose:

- optimization correspond to at least one of devirtualization (col. 3, lines 7-8 "optimization that reduce execution time (e.g., by means of call devirtualization").

Neither Sweeney nor Berg nor Bortnikov discloses optimization uses stack allocation.

However, Fitzgerald disclose in an analogous computer system optimization uses stack allocation (page 7, paragraph 6 "The stack allocation optimization improves locality...by allocating objects").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of using stack allocation during optimization as taught by Fitzgerald in the combination system as taught by Sweeney, Berg, and Bortnikov. The modification would be obvious because of one of ordinary skill in the art would be motivated to use stack allocation for optimization to improve locality and garbage collection overhead as suggested by Fitzgerald (page 7 and 8, paragraph 6).

16. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney in view of Berg, and further in view of Bortnikov.

Per claim 12:

Sweeney disclose:

- ***A method for optimizing an object oriented program*** (col. 3, lines 4-5 "The invention… used as an optimization… reduce application… eliminating unreachable methods") ***that support dynamic class loading*** (col. 2, lines 4-7 " Dynamic loading … a class with a specified name, after… instances of… class… created"), ***the program comprising a plurality of objects*** (col. 11, lines 63-64 "plurality of components comprises classes"), ***each class belonging to at least one class*** (col. 9, line 59 "dynamically load class M, create an object of type M")
- ***identifying a set A of classes in the program*** (col. 8, line 29 "identifying a set I of classes")
- ***adding an optimization to the program*** (Abstract, "whole-program optimizations")

Sweeney does not explicitly disclose wherein each of the classes within the set A is capable of during execution of the program, causing loading of a class outside of the set A.

However, Berg discloses in an analogous computer system class loader loads a class from different domains at runtime (page 249, 1st paragraph "A class loader loads classes from one or more domains… codesources, typically single codesource").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of loading a class from outside of a class at runtime as taught by Berg into the method of analyzing the of a program as taught by Sweeney. The modification would be obvious because of one of ordinary skill in the art would be motivated load the class from within the class which provide powerful JAVA platform to install software components at runtime.

Neither Sweeney nor Berg disclose validity and triggering of an optimization program.

However, Bortnikov disclose in an analogous computer system validity and execution of optimization (col. 2, lines 62-64 "optimization mechanism that optimizes using any available valid procedure").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of valid optimization of a program as taught by Bortnikov into the combination system as taught by Sweeney, and Berg. The modification would be obvious because of one of ordinary skill in the art would be motivated to have validate optimization program for optimization of the program to efficiently run the system as suggested by Bortnikov (col. 3, lines 21-36).

17. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg, Bortnikov in view of Kwong.

Per claim 13:

The rejection of claim 12 is incorporated, and further, neither Sweeney nor Berg nor Bortnikov discloses performing static analysis in identifying the first and second set.

However, Kwong discloses in an analogous computer system performing static analysis (col. 8, lines 27-28 "perform static and dynamic analysis of the program's performance").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of performing static analysis to a program as taught by Kwong into the method of analyzing the program as taught by the combination system of Sweeney, Berg, and Bortnikov. The modification would be obvious because of one of ordinary skill in the art would be motivated to perform the static analysis of the program to get the optimized code as taught by Kwong (col. 2, lines 20-34)

18. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg, Bortnikov in view of Fitzgerald.

Per claim 14:

The rejection of claim 12 is incorporated, and further, Sweeney disclose:

- ***optimization correspond to at least one of devirtualization*** (col. 3, lines 7-8 "optimization that reduce execution time (e.g., by means of call devirtualization").

The feature of optimization using stack allocation register would be obvious for the reasons set forth above in the rejection of claim 10.

19. Claims 15, 24, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney in view of Berg and further in view of Bortnikov.

Per claims 15 and 24:

Sweeney disclose:

- A method for optimizing an object oriented program (col. 3, lines 4-5 "The invention… used as an optimization… reduce application… eliminating unreachable methods") ***that support dynamic class loading*** (col. 2, lines 4-7 " Dynamic loading … a class with a specified name, after… instances of… class… created"), ***the program comprising a plurality of objects*** (col. 11, lines 63-64 "plurality of components comprises classes"), ***each class belonging to at least one class*** (col. 9, line 59 "dynamically load class M, create an object of type M")

- identify a first set of method calls (col. 8, line 27 "identifying a set M of methods… calls") ***belonging to the classes in the set A that*** (col. 8, lines 37-38 "the data identifying the set R is initialized to identify a set of initially reachable methods (such as an application's main method)"), ***during execution of the program*** (col. 12, lines 2-3 "during… execution of the program"), ***are capable of calling only methods belonging to a class within the set A*** (col. 12, lines 27-29 "components is satisfied if at least one component in the first set of components")

- for a given method call (col. 7, line 28 "For a given call… methods") ***in the second set*** (col. 6, line 38 "second set of component")

Sweeney does not explicitly disclose wherein each of the classes within the set A is capable of during execution of the program, causing loading of a class outside of the set A.

However, Berg discloses in an analogous computer system class loader loads a class from different domains a t runtime (page 249, 1st paragraph "A class loader loads classes from one or more domains… codesources, typically single codesource").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of loading a class from outside of a class at runtime as taught by Berg into the method of analyzing the of a program as taught by Sweeney. The modification would be obvious because of one of ordinary skill in the art would be motivated load the class from within the class which provide powerful JAVA platform to install software components at runtime.

Neither Sweeney nor Berg disclose validity and triggering of an optimization program.

However, Bortnikov disclose in an analogous computer system validity and execution of optimization (col. 2, lines 62-64 "optimization mechanism that optimizes using any available valid procedure"). It is inherent to execute the program for optimization for optimization mechanism.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of valid optimization of a program as taught by Bortnikov into the combination system as taught by Sweeney and Berg. The modification would be obvious because of one of ordinary skill in the art would be motivated to have validate optimization program for optimization of the program to efficiently run the system as suggested by Bortnikov (col. 3, lines 21-36).

Claim 29 is the product claim corresponding to method claims 12 and 15 respectively, and rejected under the same reason set forth in connection of the rejection of claims 12 and 15 respectively, above.

20. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg, Bortnikov in view of US Patent No. 5,987,256 to Wu et al., hereinafter called Wu.

Per claim 16:

The rejection of claim 15 is incorporated, and further, neither Sweeney nor Berg nor Bortnikov disclose the environment is off-line.

However, Wu disclose in an analogous computer system the off-line environment for producing a compiled format (col. 19, lines 47-49 "off-line fill feature HTML and JAVA processing is provided for a run time environment on a very thin client such as a VCD/DVD player").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of producing compiled format on an off-line environment as taught by Wu into the combination system of optimizing code as taught by Sweeney, Berg and Bortnikov. The modification would be obvious because of one of ordinary skill in the art would be motivated to use off-line environment in optimizing the code so that it would not be interfered by users/machines.

21. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg, Bortnikov in view of Fast Interprocedural Class Analysis, published in 1998by DeFouw et al., hereinafter called DeFouw.

Per claim 17:

The rejection of claim 15 is incorporated, and further, neither Sweeney nor Berg nor Bortnikov disclose using parametric data flow analysis.

However, DeFouw discloses in an analogous computer system optimization performed using parametric data flow analysis (page 11, paragraph 7, "Parameterized Analysis Algorithm Pseudocode for our general algorithm for interprocedural class analysis appears in Figures 7 and 8. The core of the algorithm performs propagation of classes through the dataflow graph. During the propagation phase, each supernode maintains an associated bag of classes that have reached the supernode but have not yet been processed by the supernode").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of using parametric data flow analysis as taught by DeFouw into the combination system of optimizing code as taught by Sweeney, Berg and Bortnikov. The modification would be obvious because of one of ordinary skill in the art would be motivated to use parametric data flow analysis to improve the application speed and compactness as taught by DeFouw (page 32, paragraph 2-3).

22. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg, Bortnikov in view of Fitzgerald.

Per claim 18:

The rejection of claim 15 is incorporated, and further, Sweeney disclose:

- optimization correspond to at least one of devirtualization (col. 3, lines 7-8 "optimization that reduce execution time (e.g., by means of call devirtualization").

Neither Sweeney nor Berg nor Bortnikov discloses optimization uses stack allocation.

However, Fitzgerald disclose in an analogous computer system optimization uses stack allocation (page 7, paragraph 6 "The stack allocation optimization improves locality…by allocating objects").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of using stack allocation during optimization as taught by Fitzgerald in the combination system as taught by Sweeney, Berg, and Bortnikov. The modification would be obvious because of one of ordinary skill in the art would be motivated to use stack allocation for optimization to improve locality and garbage collection overhead as suggested by Fitzgerald (page 7 and 8, paragraph 6).

23. Claims 19-23 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sweeney, Berg, Bortnikov in view of Call graph construction in object-oriented language, published in 1997 by Grove et al., hereinafter called Grove.

Per claims 19, 20, and 25:

The rejection of claim 15 is incorporated, and further, neither Sweeney nor Berg nor Bortnikov discloses a reference variable pointing to the class sets, constructing class and call graph.

However, Grove disclose in an analogous computer system call graph construction in Object Oriented Language which uses the feature as claimed (page 1, Abstract, "constructing an accurate program call graph" and page 2, section 2.1"sets of classes represent the possible classes of values stored in the corresponding variable at run-time… whose "local variables" correspond to the global variables" and "… record sets of… classes for each instance variable" and page 3, paragraph 1 "different instantiation sites of a class as leading to distinct (analysis-time) classes with distinct variable contours").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of call graph as taught by Grove into the into the method of optimizing the program as taught by the combination system of Sweeney, Berg, and Bortnikov. The modification would be obvious because of one of ordinary skill in the art would be motivated to use program call graph to increase the performance of the application as suggested by Grove (page 1, Abstract).

Per claims 21, 22, and 23:

The rejection of claim 20 is incorporated, and further, Sweeney disclose:

- ***adding a bit*** (col. 9, line 60 "assign the value '10'… integer field x") ***in a class table*** (col. 9, line 56 "small class libray") ***corresponding to a given class and setting the bit to a first predefined value when the given class is within the set A*** (col. 10, lines 27-45 "initial value… sets are… instructions in the… method… are scanned… contains virtual calls… a direct call… an initiation of class C… data identifying … is added to the data identifying set M" and col. 8, lines 64-67 "Steps… perform… new classes are added to set…").

Conclusion

24. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

The following patent is cited to further show the state of the art with respect to optimization of object oriented program.

US Patent No. 5,535,391 to Hejlsberg et al.

US Patent No. 6,202,205 to Saboff et al.

US Patent No. 5,970,249 to Holzle et al.

Any inquiry concerning this communication or earlier communication from the examiner should be directed to Satish Rampuria whose telephone number is 703-305-8891.

The examiner can normally be reached on Monday-Friday from 8:30 A. M. to 5:00 P.M. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor Kakali Chaki can be reached at 703-305-9662. The fax number for this group is 703-872-9306. An inquiry of general nature or relating to the status of this application or proceeding should be directed to the group receptionist whose telephone number is 703-305-3900.

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Satish S. Rampuria

Patent Examiner

Art Unit 2124

04/05/04